INTERNATIONAL STANDARD

ISO/IEC 14496-22

Third edition
2015-10-01

AMENDMENT 2
2017-08

Information technology — Coding of audio-visual objects —

Part 22: Open Font Format

AMENDMENT 2: Updated text layout features and implementations

Technologies de l'information — Codage des objets audiovisuels —

Partie 22: Format de police de caractères ouvert

AMENDEMENT 2: Mise à jour de l'introduction et des caractéristiques de mise en page du texte

Reference number
ISO/IEC 14496-22:2015/Amd.2:2017(E)

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see the following URL: www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

Information technology — Coding of audio-visual objects —

Part 22: Open Font Format

AMENDMENT 2: Updated text layout features and implementations

Normative references

Replace the reference to the Unicode Standard with the following text:

Unicode 9.0, < http://www.unicode.org/versions/Unicode9.0.0/>

5.2.7.3

At the end of the "Description" field, add the following text:

Only values from 1 to 1000 are valid.

In the "Comments" field, add the following text preceding the table:

There may be legacy platform limitations on certain usWeightClass values. The following are commonly set values:

5.5.1

In "SVG Document Index Entry", replace the second before the last paragraph with the following:

While SVG 1.1 requires [16] in addition to plain text encoding that conforming SVG implementations must correctly support gzip-encoded [RFC 1952] and deflate-encoded [RFC 1951] data streams, this specification requires that the SVG documents be either plain-text or gzip-encoded [RFC 1952]. The encoding of the (uncompressed) SVG document must be UTF-8. In both cases, svgDocLength encodes the length of the encoded data, not the decoded document.

5.5.4

Replace the first paragraph with the following:

The glyph descriptions in the SVG documents are considered to be the SVG versions of the glyphs with the corresponding IDs in the CFF or glyf table. They are designed on an em specified in the head table's unitsPerEm field, as with CFF and TrueType glyphs. SVG glyph definitions will be in SVG's own y-down coordinate system, upright, with the default baseline at y=0. For example, the top of a capital letter may be at y=−800, and the bottom at y=0 (see Examples section below). It is the font engine's responsibility to translate this to the coordinate system of the rest of the OT tables and the coordinate system of the graphics environment.

5.5.5

Replace the first sentence of the second paragraph with the following:

> The font engine must apply the following user agent style sheet (or implement its functional equivalent) to SVG documents processed from the SVG table:

In "Security considerations and other glyph rendering restrictions", replace the second paragraph with the following:

> These requirements correspond to the "secure animated" and "secure static" processing modes that the SVG Integration document [17] requires font documents to be run in.

In "Security considerations and other glyph rendering restrictions", replace the third (last) paragraph with the following:

> In addition, any SVG <text> and <foreignObject> elements within a glyph description must be ignored and not rendered (see the corresponding rules in the User Agent style sheet above).

Replace the fourth and fifth paragraphs with the following:

> The font engine must support at least version 1.1 of the SVG specification (exceptions are noted in the section on glyph rendering restrictions). The version attribute in the <svg> element is present in the SVG 1.1 and 1.2 specifications, but not in SVG 2. Thus, the SVG document may not always have a version field specified. Given this approach to versioning in SVG, and given that not all implementations may support all of SVG (whether 1.1 or 2), font designers should restrict their SVG, as a practical matter, to whatever is supported by SVG-in-OT implementations they care about. Targeting the capabilities of SVG 1.1 would be the approach most likely to result in cross-implementation consistency.

5.5

Add new subclause “5.5.6 SVG glyph examples” with the following content:

> SVG glyph descriptions must be defined in SVG’s own y-down coordinate system, upright, with the default baseline at y=0. It is *always* the font engine’s responsibility to translate this into the coordinate system of the rest of the OFF font rendering environment.
>
> The SVG code in these examples is presented exactly as could be used in the SVG documents of an OFF font with SVG glyph outlines. The code is not optimized for brevity.

Example: Glyph specified directly in expected position


```
<svg id="glyph7" version="1.1" xmlns="http://www.w3.org/2000/svg">
  <defs>
    <linearGradient id="grad" x1="0%" y1="0%" x2="0%" y2="100%">
      <stop offset="0%" stop-color="darkblue" stop-opacity="1" />
      <stop offset="100%" stop-color="#00aab3" stop-opacity="1" />
    </linearGradient>
  </defs>
  <rect x="100" y="-430" width="200" height="430" fill="url(#grad)" />
  <rect x="100" y="-635" width="200" height="135" fill="darkblue" />
</svg>
```

In this example, the letter “i” is drawn directly in the +x –y quadrant of the SVG coordinate system, upright, with its baseline on the x axis, exactly where the OFF font engine expects it to be.

Example: Glyph shifted up with viewBox

```
<svg id="glyph7" version="1.1" xmlns="http://www.w3.org/2000/svg" viewBox="0 1000 1000
1000">
  <defs>
    <linearGradient id="grad" x1="0%" y1="0%" x2="0%" y2="100%">
      <stop offset="0%" stop-color="darkblue" stop-opacity="1" />
      <stop offset="100%" stop-color="#00aab3" stop-opacity="1" />
    </linearGradient>
  </defs>
  <rect x="100" y="570" width="200" height="430" fill="url(#grad)" />
  <rect x="100" y="365" width="200" height="135" fill="darkblue" />
</svg>
```


In this example, the glyph description of the letter “i” is first specified in the +x +y quadrant of the SVG coordinate system, upright, with its baseline along y=1000 in the SVG coordinate system. (This may be the natural way the SVG illustrating software positioned it.) A viewBox in the <svg> element is then used to shift it upwards by 1000 units, to end up in the position where the OFF font engine expects it to be.

The diagram is the same as in the above example.

Example: Common elements shared across glyphs in same SVG doc

```
<svg version="1.1" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/
xlink">
  <defs>
    <linearGradient id="grad" x1="0%" y1="0%" x2="0%" y2="100%">
      <stop offset="0%" stop-color="darkblue" stop-opacity="1" />
      <stop offset="100%" stop-color="#00aab3" stop-opacity="1" />
    </linearGradient>
    <g id="i-base">
      <rect x="100" y="570" width="200" height="430" fill="url(#grad)" />
```

```
    </g>
  </defs>
  <g id="glyph2" transform="translate(0,-1000)">
    <use xlink:href="#i-base" />
  </g>
  <g id="glyph13" transform="translate(0,-1000)">
    <use xlink:href="#i-base" />
    <rect x="100" y="365" width="200" height="135" fill="darkblue" />
  </g>
  <g id="glyph14" transform="translate(0,-1000)">
    <use xlink:href="#i-base" />
    <polygon fill="darkblue" points="120,500 280,500 435,342 208,342" />
  </g>
</svg>
```

In this example, the base of the letter 'i' is shared across three glyphs, and has identifier “i-base” in the <defs> section. It represents the dotless 'i' in glyph ID 2. Glyph ID 13 adds a dot on top. Glyph ID 14 adds an acute accent on top. The diagram above shows glyph IDs 2, 13, and 14, from left to right.

Note that glyph IDs 3-12 can be defined in one or more separate SVG docs, and still allow glyph IDs 2, 13, and 14 to share the same SVG doc. For example:

SVG Document Index: numEntries=5

...

entries[2]: { startGlyphID = 2, endGlyphID = 2, svgDocOffset/Length point to svgDoc0 }

entries[3]: { startGlyphID = 3, endGlyphID = 12, svgDocOffset/Length point to svgDoc1 }

entries[4]: { startGlyphID = 13, endGlyphID = 14, svgDocOffset/Length point to svgDoc0 }

Example: Specifying current text color in glyphs

```
<svg id="glyph7" version="1.1" xmlns="http://www.w3.org/2000/svg" viewBox="0 1000 1000
1000">
  <defs>
    <linearGradient id="grad" x1="0%" y1="0%" x2="0%" y2="100%">
      <stop offset="0%" stop-color="darkblue" stop-opacity="1" />
      <stop offset="100%" stop-color="#00aab3" stop-opacity="1" />
    </linearGradient>
  </defs>
  <rect x="100" y="570" width="200" height="430" fill="url(#grad)" />
  <rect x="100" y="365" width="200" height="135" fill="context-fill" />
</svg>
```

Here the “darkblue” color of the dot above the “i” in the “Glyph shifted up with viewBox” example is replaced by “context-fill”. The diagram above shows the glyph when the fill color of the context element (i.e. the text color) is set to black (left) and red (right).

Example: Specifying color palette variables in glyphs

```
<svg id="glyph7" version="1.1" xmlns="http://www.w3.org/2000/svg" viewBox="0 1000 1000
1000">
  <defs>
    <linearGradient id="grad" x1="0%" y1="0%" x2="0%" y2="100%">
      <stop offset="0%" stop-color="var(--color0,darkblue)" stop-opacity="1" />
      <stop offset="100%" stop-color="var(--color1,#00aab3)" stop-opacity="1" />
    </linearGradient>
  </defs>
  <rect x="100" y="570" width="200" height="430" fill="url(#grad)" />
  <rect x="100" y="365" width="200" height="135" fill="darkblue" />
</svg>
```


This example is the duplicate of the “Glyph shifted up with viewBox” example, but with the stop colors of the linear gradient controlled by color variables --color0 and --color1, which are provided by the font engine to the SVG renderer via a user agent style sheet (or its functional equivalent).

The color palettes (CPAL) table in this font specifies two palettes, each with two color entries. Here is a description of the CPAL palettes, with alpha assumed to be 0xFF for all colors:

palette[0]: { darkblue, #00aab3 }

palette[1]: { purple, orchid }

The first item in the diagram above shows the first color palette applied to the glyph, which is done by the font engine passing the following user agent style sheet to the SVG renderer:

:root {

--color0: darkblue;

--color1: #00aab3;

}

The second item in the diagram shows the second color palette applied to the glyph, using the style sheet:

:root {

--color0: purple;

--color1: orchid;

}

Note that the dot is still dark blue, since this is hard coded in the glyph description and not controlled by a color variable.

The last item in the diagram shows the following user-selected colors applied to the glyph via the color variables:

:root {

--color0: red;

--color1: orange;

}

If --color0 and --color1 are not defined by the font engine, however, then the default values provided in the stop-colors (darkblue and #00aab3, respectively) are used. Note that these are in fact the same colors as in the first (default) CPAL color palette, which means the glyph will render as in the first item in the diagram. This way, the glyph renders with the same colors by default, whether or not the font engine supports the CPAL.

Example: Embedding a PNG in an SVG glyph

```
<svg id="glyph2" version="1.1" xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink" viewBox="0 1000 1000 1000">
  <image x="100" y="365" width="200" height="635"
  xlink:href="
iVBORw0KGgoAAAANSUhEUgAAAMgAAAJ7CAYAAACmmd5sAAAFZklEQVR42u3XsQ3D
MBAEQUpw9ypahrMPGGwiwcFMCQQW9zzWuu4FbJ2eAAQCAgGBgEBAICAQEAgIBAQC
CAQEAgIBgYBAQCAgEBAICAQEAggEBAICAYGAQEAgIBAQCAgEEAgIBAQCAgGBgEBA
ICAQEAgIBBAICAQEAgIBgYBAQCAgEBAIIBAQCAgEBAICAYGAQEAgIBAQCCAQEAgI
BAQCAgGBgEBAICAQQCAgEBAICAQEAgIBgYBAQCAgEEAgIBAQCAgEBAICAYGAQEAg
IBBPAAIBgYBAQCAgEBAICAQEAgIBBAICAYGAQEAgIBAQCAgEBAICAQQCAgGBgEBA
ICAQEAgIBAQCCAQEAgIBgYBAQCAgEBAICAQEAggEBAICAYGAQEAgIBAQCAgEAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAA4DHHWtftGWDv80sE2Ds9AQgEBAL+IPBuIAoBJxYIBAQCPukgEHBigUBAIOAP
AlgQiAtiQsCCgEDAJx0sCFgQsCAgEHBigQUB5oKYELAgIBDwSQcLAhYELAgIBJxY
YEEACwItEIWAEwucWGBBwIKABQGBgBMLLAhYEMCCQFwQEwJOLHBigQUBCwICAScW
WBCwIGBBAIFAPbHcWGBBwCcdLAgIBJxYYEHAgoAFAYEA88RyY4EFAZ90sCAgEBAI
+IOAQMCJBQIBBALxD+ITAj7p4MQCgYBAwB8EBAJOLBAICATwB4EYiELAiQUCAYGA
TzoIBJxYIBAQCPiDABYE4oKYELAgIBDwSQcLAhYELAgIBJxYYEGAuSAmBCwICAR8
0sGCgAUBCwICAScWWBDAgkALRCHgxAInFlgQsCBgQUAg4MQCCwIWBLAgEBfEhIAT
C5xYYEHAgoBAwIkFFgQsCFgQQCBQTyw3FlgQ8EkHCwICAScWWBCwIGBBQCDAPLHc
WGBBwCcdLAgIBAQC/iAgEHBigUAAgUD8g/iEgE86OLFAICAQ8AcBgYATCwQCAgH8
```